Possession as Linear Knowledge

Frank Pfenning

[with Deepak Garg, Henry DeYoung,
and Michael Ashley-Rollman]

Department of Computer Science
Carnegie Mellon University

3rd International Workshop on
Logics, Agents, and Mobility (LAM)
July 15, 2010

Understanding Distributed Systems

- Goals

Understanding Distributed Systems

- Goals
  - Logical specification of distributed authorization policies

Understanding Distributed Systems

- Goals
  - Logical specification of distributed authorization policies
  - Reliable enforcement of such high-level policies

Understanding Distributed Systems

- Goals
  - Logical specification of distributed authorization policies
  - Reliable enforcement of such high-level policies
  - Mechanized reasoning about consequences of policies:
    - Evolution of system state
    - Principals' knowledge (information)
    - Principals' possessions (consumable resources)

Understanding Distributed Systems

- Goals
  - Logical specification of distributed authorization policies
  - Reliable enforcement of such high-level policies
  - Mechanized reasoning about consequences of policies:
    - Evolution of system state
    - Principals' knowledge (information)
    - Principals' possessions (consumable resources)
- Approach: linear epistemic logic

Understanding Distributed Systems

- Goals
  - Logical specification of distributed authorization policies
  - Reliable enforcement of such high-level policies
  - Mechanized reasoning about consequences of policies:
    - Evolution of system state
    - Principals' knowledge (information)
    - Principals' possessions (consumable resources)
- Approach: linear epistemic logic
- Examples
  - Documents in the intelligence community of the US
  - Course management
  - Monetary instruments
  - File system

Outline

1. Background: Proof-Carrying Authorization
2. Logical Foundations
   1. Resources (linear logic)
   2. Possessions (linear epistemic logic)
   3. Effects (linear lax logic)
   4. From axioms to inference rules via focusing
   5. Persistent truth and knowledge (epistemic logic)
3. Policy Consequences
   1. State invariants
   2. Proving metatheorems
4. Speculation: linear epistemic logic programming

Background: Authorization Logic

- Logic for distributed authorization
  - Authorization policy is stated as a logical theory T
  - Principal K can perform operation O if authorization proposition $\mathrm{may}(K, O)$ is true in T
  - The proof embodies the reason why action should be permitted
- Core: "K **says** A" for principal K and proposition A
  - Family of K-indexed modal operators
  - Precise definition not important for this talk

Background: Proof-Carrying Authorization

- Enforcement architecture for access control
- "K **says** A" can be realized in two ways
  - Proposition "A" digitally signed by K
  - Explicit proof using logical inference
- Policy theory consists of signed "K **says** A"
- Reference monitor grants access if formal proof object "$M : K$ **says** $\mathsf{may}(L, O)$" is correct (for resource owner K)
- Core: Proof checking and certificate verification
- Examples
  - Gray (office access with smartphones)
  - Nexus (document viewer application suite)
  - PCFS (proof-carrying file system)

Example: A Versioned File System

Key to Syntax
$\langle K \rangle A =$ "K **says** A"

Principals K, L:	$fs, \ldots$
Operations O:	create, on(F, A)
Actions A:	read, write(s), delete
Propositions:	$\langle fs \rangle \mathrm{user}(K)$
	$\langle fs \rangle \mathrm{owns}(K, F)$
	$\langle fs \rangle \mathrm{may}(L, O), \langle K \rangle \mathrm{may}(L, O)$

Sample policy, file system

$$
\begin{aligned}
\mathrm{create} &: \langle fs \rangle (\mathrm{user}(K) \supset \mathrm{may}(K, \mathrm{create})) \\
\mathrm{delegate} &: \langle fs \rangle (\mathrm{owns}(K, F) \wedge \langle K \rangle \mathrm{may}(L, \mathrm{on}(F, A)) \\
&\qquad\qquad \supset \mathrm{may}(L, \mathrm{on}(F, A)))
\end{aligned}
$$

Example: Distributed Policy

Key to Syntax
$\langle K \rangle A =$ "K **says** A"

Sample policy, Alice

$$\langle \mathrm{alice} \rangle (\langle fs \rangle \mathrm{owns}(\mathrm{alice}, F) \supset \mathrm{may}(\mathrm{alice}, \mathrm{on}(F, A)))$$

$$\langle \mathrm{alice} \rangle (\mathrm{friend}(K, \mathrm{alice})) \supset \mathrm{may}(K, \mathrm{on}(\mathrm{embarassing.jpg}, \mathrm{read}))$$

$$\langle \mathrm{alice} \rangle (\mathrm{friend}(K, \mathrm{alice}) \wedge \langle K \rangle \mathrm{friend}(L, K) \supset \mathrm{may}(L, \mathrm{on}(\mathrm{fun.jpg}, \mathrm{read})))$$

Background: Single-Use Authorization

- Access to or with consumable resources
  - "K **says** pay($K, L, \$50$)"
  - "netflix **says** may(L, playmovie(3))"
- Core: linear authorization logic
- Enforcement
  - Linear digitally signed certificates
  - Linear proof checking
  - Reference counting in resource monitor
- Atomicity: multi-party contract signing

Semantics

- Capture consequences of authorization policy
  - Information flow: what knowledge may principals gain?
  - Accounting: what possessions may principals obtain or relinquish?
- Which states of knowledge and possession can be reached?
- Verify desirable semantic consequences
  - "To learn the contents of a file, one must have read or write access"
  - "Banking machines fees for a single transaction will be no more than $2"
  - "Every valid electronic vote will be counted"
- Caveat: we stay within the level of abstraction of the semantic description

Example: File System State

Key to Syntax
$\langle K \rangle A$ = "K **says** A"
$[K]A$ = "K **has** A"
$[\![K]\!]A$ = "K **knows** A"
$\{A\}$ = "A, with effect"

Command:	$\langle K \rangle \mathrm{do}(K, O)$	linear
Version:	$[K]\mathrm{current}(F, V)$	possession – linear
Contents:	$[\![K]\!]\mathrm{contains}(F, V, S)$	knowledge – persistent

- Sample rule: Creating a file

$$
\begin{aligned}
& \langle K \rangle \mathrm{do}(K, \mathrm{create}) \\
\otimes\ & \langle fs \rangle \mathrm{may}(K, \mathrm{create}) \\
\multimap\ & \{\exists f.\, \exists v. \\
& \qquad !\langle fs \rangle \mathrm{owns}(K, f) \\
& \qquad \otimes [fs]\mathrm{current}(f, v) \\
& \qquad \otimes [\![fs]\!]\mathrm{contains}(f, v, \text{""}) \\
& \qquad \otimes [\![K]\!]\mathrm{contains}(f, v, \text{""})\}
\end{aligned}
$$

Example: Reading a File

Key to Syntax
$\langle K \rangle A$ = "K **says** A"
$[K]A$ = "K **has** A"
$[\![K]\!]A$ = "K **knows** A"
$\{A\}$ = "A, with effect"

$$
\begin{array}{ll}
& \langle K \rangle \mathrm{do}(K, \mathrm{on}(F, \mathrm{read})) \\
\otimes & \langle fs \rangle \mathrm{may}(K, \mathrm{on}(F, \mathrm{read})) \\
\otimes & [fs]\mathrm{current}(F, V) \\
\otimes & [\![fs]\!]\mathrm{contents}(F, V, S) \\
\multimap & \{[fs]\mathrm{current}(F, V) \\
& \quad \otimes [\![K]\!]\mathrm{contents}(F, V, S)\}
\end{array}
$$

Example: Writing to a File

Key to Syntax
$\langle K \rangle A$ = "K **says** A"
$[K]A$ = "K **has** A"
$[\![K]\!]A$ = "K **knows** A"
$\{A\}$ = "A, with effect"

$$
\begin{array}{ll}
 & \langle K \rangle \mathrm{do}(K, \mathrm{on}(F, \mathrm{write}(S))) \\
\otimes & \langle fs \rangle \mathrm{may}(K, \mathrm{on}(F, \mathrm{write}(S))) \\
\otimes & [fs]\mathrm{current}(F, V) \\
\multimap & \{\exists v'.\, [fs]\mathrm{current}(F, v') \\
 & \quad \otimes [\![fs]\!]\mathrm{contains}(F, v', S) \\
 & \quad \otimes [\![K]\!]\mathrm{contains}(F, v', S)\}]
\end{array}
$$

Example: Deleting a File

Key to Syntax
$\langle K\rangle A$ = "K **says** A"
$[K]A$ = "K **has** A"
$[\![K]\!]A$ = "K **knows** A"
$\{A\}$ = "A, with effect"

$$
\begin{array}{ll}
 & \langle K\rangle \mathrm{do}(K, \mathrm{on}(F, \mathrm{delete})) \\
\otimes & \langle \mathit{fs}\rangle \mathrm{may}(K, \mathrm{on}(F, \mathrm{delete})) \\
\otimes & [\mathit{fs}]\mathrm{current}(F, V) \\
\multimap & \{\mathbf{1}\}
\end{array}
$$

Outline

1. Background: Proof-Carrying Authorization
2. Logical Foundations
    1. Resources (linear logic)
    2. Possessions (linear epistemic logic)
    3. Effects (linear lax logic)
    4. From axioms to inference rules via focusing
    5. Persistent truth and knowledge (epistemic logic)
3. Policy Consequences
    1. State invariants
    2. Proving metatheorems
4. Speculation: linear epistemic logic programming

Logical Foundations

- Goal: define a suitable linear logic of (authorization), possession, knowledge, and effects — linear epistemic logic
- Use such a logic
  - Logically: specifying the consequences of authorization policies
  - Metalogically: reasoning about all possible action sequences
  - Operationally: implementing (or checking implementation against) linear epistemic specification

Proof-Theoretic Semantics

- How do we define the right logic?
- The crucial role of proofs
  - Explicit evidence for authorization
  - Explicit evidence for right-to-know
  - Explicit evidence for transactions
  - Explicit traces of system evolution
- In combination with cryptographic techniques
  - Digital signatures
  - Encryption and decryption

Judgments and Propositions

$$\overbrace{\underbrace{A_1\ \mathit{res}, \ldots, A_n\ \mathit{res}}_{\substack{\Delta \\ \text{consumable resources} \\ \text{linear assumptions} \\ \text{antecedents}}} \quad \Longrightarrow \quad \underbrace{C\ \mathit{true}}_{\substack{\gamma \\ \text{goal} \\ \text{conclusion} \\ \text{succedent}}}}^{\text{linear sequent}}$$

Judgmental Principles

Identity: With resource A we can achieve goal A

$$\frac{}{A\ \mathit{res} \Longrightarrow A\ \mathit{true}}\ \mathsf{id}_A$$

Cut: If we can achieve A we can use it as a resource

$$\frac{\Delta \Longrightarrow A\ \mathit{true} \qquad \Delta', A\ \mathit{res} \Longrightarrow \gamma}{\Delta, \Delta' \Longrightarrow \gamma}\ \mathsf{cut}_A$$

- These must be admissible rules (metatheorems)
- Harmony between resources and goals

Simultaneous Conjunction $A \otimes B$

- Right rule: how to prove goal can be achieved

$$\frac{\Delta_A \Longrightarrow A \quad \Delta_B \Longrightarrow B}{\Delta_A, \Delta_B \Longrightarrow A \otimes B} \otimes R$$

- Left rule: how to use resource

$$\frac{\Delta, A, B \Longrightarrow \gamma}{\Delta, A \otimes B \Longrightarrow \gamma} \otimes L$$

(Elide *res* and *true* since clear from position)

Local Harmony

- Show how to expand

$$\dfrac{}{A \Longrightarrow A}\ \mathsf{id}_A \quad \longrightarrow_E ?$$

  using identity on subformulas of A
  - Part of proof of global identity proof by induction on A
  - Need primitive rule $P \Longrightarrow P$ for atomic P
- Show how to reduce

$$\dfrac{\begin{array}{c}\mathcal{D}\\ \Delta \Longrightarrow A\end{array} \quad \begin{array}{c}\mathcal{E}\\ \Delta', A \Longrightarrow \gamma\end{array}}{\Delta, \Delta' \Longrightarrow \gamma}\ \mathsf{cut}_A \quad \longrightarrow_R ?$$

  using cut on subformulas of A
  - Part of global cut proof by nested induction on A, $\mathcal{D}$, $\mathcal{E}$

Local Harmony for $A \otimes B$

- Identity expansion

$$\dfrac{}{A \otimes B \Longrightarrow A \otimes B}\,\mathrm{id}_{A\otimes B} \quad \longrightarrow_E \quad \dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,\mathrm{id}_A \qquad \dfrac{}{B \Longrightarrow B}\,\mathrm{id}_B}{A, B \Longrightarrow A \otimes B}\,\otimes R}{A \otimes B \Longrightarrow A \otimes B}\,\otimes L$$

Local Harmony for $A \otimes B$

- Cut reduction

$$
\dfrac{\dfrac{\dfrac{\mathcal{D}_A}{\Delta_A \Longrightarrow A} \quad \dfrac{\mathcal{D}_B}{\Delta_B \Longrightarrow B}}{\Delta_A, \Delta_B \Longrightarrow A \otimes B}\,\otimes R \qquad \dfrac{\dfrac{\mathcal{E}}{\Delta, A, B \Longrightarrow \gamma}}{\Delta, A \otimes B \Longrightarrow \gamma}\,\otimes L}{\Delta, \Delta_A, \Delta_B \Longrightarrow \gamma}\,\mathrm{cut}_{A \otimes B}
$$

$$
\longrightarrow_R \quad \dfrac{\dfrac{\mathcal{D}_B}{\Delta_B \Longrightarrow B} \qquad \dfrac{\dfrac{\mathcal{D}_A}{\Delta_A \Longrightarrow A} \quad \dfrac{\mathcal{E}}{\Delta, A, B \Longrightarrow \gamma}}{\Delta, \Delta_A, B \Longrightarrow \gamma}\,\mathrm{cut}_A}{\Delta, \Delta_A, \Delta_B \Longrightarrow \gamma}\,\mathrm{cut}_B
$$

Linear Implication $A \multimap B$

- Right rule: how to prove $A \multimap B$

$$\frac{\Delta, A \Longrightarrow B}{\Delta \Longrightarrow A \multimap B} \multimap R$$

- Left rule: how to use $A \multimap B$

$$\frac{\Delta_A \Longrightarrow A \quad \Delta_B, B \Longrightarrow \gamma}{\Delta_A, \Delta_B, A \multimap B \Longrightarrow \gamma} \multimap L$$

Identity Expansion for $A \multimap B$

$$\dfrac{}{A \multimap B \Longrightarrow A \multimap B}\ \mathrm{id}_{A \multimap B} \quad \longrightarrow_E \quad \dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\ \mathrm{id}_A \qquad \dfrac{}{B \Longrightarrow B}\ \mathrm{id}_B}{A \multimap B, A \Longrightarrow B}\ {\multimap}L}{A \multimap B \Longrightarrow A \multimap B}\ {\multimap}R$$

Cut Reduction for $A \multimap B$

$$
\dfrac{\dfrac{\overset{\mathcal{D}}{\Delta, A \Longrightarrow B}}{\Delta \Longrightarrow A \multimap B}\ {\multimap}R \qquad \dfrac{\overset{\mathcal{E}_A}{\Delta_A \Longrightarrow A} \quad \overset{\mathcal{E}_B}{\Delta_B, B \Longrightarrow \gamma}}{\Delta_A, \Delta_B, A \multimap B \Longrightarrow \gamma}\ {\multimap}L}{\Delta, \Delta_A, \Delta_B \Longrightarrow \gamma}\ \mathrm{cut}_{A \multimap B}
$$

$$
\longrightarrow_R \quad \dfrac{\dfrac{\overset{\mathcal{E}_A}{\Delta_A \Longrightarrow A} \quad \overset{\mathcal{D}}{\Delta, A \Longrightarrow B}}{\Delta, \Delta_A \Longrightarrow B}\ \mathrm{cut}_A \qquad \overset{\mathcal{E}_B}{\Delta_B, B \Longrightarrow \gamma}}{\Delta, \Delta_A, \Delta_B \Longrightarrow \gamma}\ \mathrm{cut}_B
$$

Unit Resource $\mathbf{1}$

$$\frac{}{\bullet \Longrightarrow \mathbf{1}}\ \mathbf{1}R \qquad \frac{\Delta \Longrightarrow \gamma}{\Delta, \mathbf{1} \Longrightarrow \gamma}\ \mathbf{1}L$$

$$\frac{}{\mathbf{1} \Longrightarrow \mathbf{1}}\ \mathrm{id}_{\mathbf{1}} \quad \longrightarrow_E \quad \dfrac{\dfrac{}{\bullet \Longrightarrow \mathbf{1}}\ \mathbf{1}R}{\mathbf{1} \Longrightarrow \mathbf{1}}\ \mathbf{1}L$$

$$\dfrac{\dfrac{}{\bullet \Longrightarrow \mathbf{1}}\ \mathbf{1}R \quad \dfrac{\begin{array}{c}\mathcal{E}\\ \Delta \Longrightarrow \gamma\end{array}}{\Delta, \mathbf{1} \Longrightarrow \gamma}\ \mathbf{1}L}{\Delta \Longrightarrow \gamma}\ \mathrm{cut}_{\mathbf{1}} \quad \longrightarrow_R \quad \begin{array}{c}\mathcal{E}\\ \Delta \Longrightarrow \gamma\end{array}$$

"$\bullet$" denotes no resources

Example: Resources

- Example: $\$, \$, \$, (\$ \otimes \$ \multimap \text{coffee}) \Longrightarrow \text{coffee} \otimes \$$

$$\dfrac{\dfrac{\dfrac{}{\$ \Longrightarrow \$}\,\text{id} \quad \dfrac{}{\$ \Longrightarrow \$}\,\text{id}}{\$, \$ \Longrightarrow \$ \otimes \$}\,\otimes R \qquad \dfrac{\dfrac{}{\text{coffee} \Longrightarrow \text{coffee}}\,\text{id} \quad \dfrac{}{\$ \Longrightarrow \$}\,\text{id}}{\$, \text{coffee} \Longrightarrow \text{coffee} \otimes \$}\,\otimes R}{\$, \$, \$, (\$ \otimes \$ \multimap \text{coffee}) \Longrightarrow \text{coffee} \otimes \$}\,\multimap L$$

- In a proof, all resources have to be used exactly once

$$\$, \$, \$, (\$ \otimes \$ \multimap \text{coffee}) \not\Longrightarrow \text{coffee}$$

$$\$, (\$ \otimes \$ \multimap \text{coffee}) \Longrightarrow \$ \multimap \text{coffee}$$

- $\$ \otimes \$ \multimap$ coffee should be an axiom that we can use as often as we want

Example: Possession

- Previous example is imprecise: who has the dollars and who has the coffee? More precise (tdo = Tazza D'Oro)

  $$[fp]\$ \otimes [fp]\$ \otimes [tdo]\mathrm{beans} \multimap [fp]\mathrm{coffee} \otimes [tdo]\$ \otimes [tdo]\$$$

- Need possession modality $[K]A$ ("K **has** A")

Possession as a Judgment

- New judgment: K **has** A (used as assumption)
- Judgmental rule: K can relinquish possession

$$\frac{\Delta, A\ \mathit{res} \Longrightarrow \gamma}{\Delta, K\ \textbf{has}\ A \Longrightarrow \gamma}\ \textbf{has}_L$$

- K cannot gain possession (arbitrarily)
- Judgmental definition: (always silently expanded on right)

$$\left[\ \frac{\Delta|_K \Longrightarrow A\ \mathit{true}}{\Delta|_K \Longrightarrow K\ \textbf{has}\ A}\ \textbf{has}_R\ \right]$$

(The rule $\textbf{has}_R$ is drawn with a double inference line.)

- $\Delta|_K$ only has antecedents of the form "K **has** A"

Identity and Cut

- No new identity principle

$$\dfrac{\dfrac{\overline{A \Longrightarrow A}\;\mathsf{id}}{K\ \mathbf{has}\ A \Longrightarrow A}\;\mathbf{has}_L}{K\ \mathbf{has}\ A \Longrightarrow K\ \mathbf{has}\ A}\;\mathbf{has}_R$$

- Derived cut principle

$$\dfrac{\dfrac{\Delta|_K \Longrightarrow A}{\Delta|_K \Longrightarrow K\ \mathbf{has}\ A}\;\mathbf{has}_R \qquad \Delta', K\ \mathbf{has}\ A \Longrightarrow \gamma}{\Delta|_K, \Delta' \Longrightarrow \gamma}\;\mathsf{cut}_{\mathbf{has}}$$

Possession as a Proposition

- Internalize K **has** A judgment as a proposition $[K]A$

$$\frac{\Delta|_K \Longrightarrow A}{\Delta|_K \Longrightarrow [K]A}\,[\,]R \qquad \frac{\Delta, K\ \mathbf{has}\ A \Longrightarrow \gamma}{\Delta, [K]A \Longrightarrow \gamma}\,[\,]L$$

Identity Expansion for Possession

$$\dfrac{}{[K]A \Longrightarrow [K]A}\,\mathsf{id}_{[K]A} \quad\longrightarrow_E\quad \dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,\mathsf{id}}{K\ \mathbf{has}\ A \Longrightarrow A}\,\mathbf{has}_L}{K\ \mathbf{has}\ A \Longrightarrow [K]A}\,[\,]R}{[K]A \Longrightarrow [K]A}\,[\,]L$$

Cut Reduction for Possession

$$\dfrac{\dfrac{\begin{array}{c}\mathcal{D}\\ \Delta|_K \Longrightarrow A\end{array}}{\Delta|_K \Longrightarrow [K]A}\,[\,]R \quad \dfrac{\begin{array}{c}\mathcal{E}\\ \Delta', K\ \mathbf{has}\ A \Longrightarrow \gamma\end{array}}{\Delta', [K]A \Longrightarrow \gamma}\,[\,]L}{\Delta|_K, \Delta' \Longrightarrow \gamma}\,\mathsf{cut}_{[K]A}$$

$$\longrightarrow_R \quad \dfrac{\begin{array}{c}\mathcal{D}\\ \Delta|_K \Longrightarrow A\end{array} \quad \begin{array}{c}\mathcal{E}\\ \Delta', K\ \mathbf{has}\ A \Longrightarrow \gamma\end{array}}{\Delta|_K, \Delta' \Longrightarrow \gamma}\,\mathsf{cut}_{K\mathbf{has}A}$$

Axiomatics

- Axioms like Intuitionistic S4, but linear

$$\begin{array}{ll} \vdash [K](A \multimap B) \multimap ([K]A \multimap [K]B) & (K^{\Box}) \\ \vdash [K]A \multimap [K][K]A & (4^{\Box}) \\ \vdash [K]A \multimap A & (T^{\Box}) \end{array}$$

- Rule of necessitation

$$\frac{\vdash A}{\vdash [K]A}\ (\text{nec})$$

Outline

1. Background: Proof-Carrying Authorization
2. Logical Foundations
    1. Resources (linear logic)
    2. Possessions (linear epistemic logic)
    3. Effects (linear lax logic)
    4. From axioms to inference rules via focusing
    5. Persistent truth and knowledge (epistemic logic)
3. Policy Consequences
    1. State invariants
    2. Proving metatheorems
4. Speculation: linear epistemic logic programming

The Effect Monad

- Applying rules such as

  $$[fp]\$ \otimes [fp]\$ \otimes [tdo]\mathrm{beans} \multimap [fp]\mathrm{coffee} \otimes [tdo]\$ \otimes [tdo]\$$$

  represent a change of state
- Proofs of authorizations such as $\langle fs \rangle \mathrm{may}(K, \mathrm{on}(F, \mathrm{read}))$ do not involve a change of state
- Isolate changes in an effect monad
- Logically, this is a lax modality $\{A\}$
- Rewrite above as

  $$[fp]\$ \otimes [fp]\$ \otimes [tdo]\mathrm{beans} \multimap \{[fp]\mathrm{coffee} \otimes [tdo]\$ \otimes [tdo]\$\}$$

Lax Judgment

- New judgment $A\ lax$ (A is true with effect)
- Judgmental rule: truth entails lax truth

$$\frac{\Delta \Longrightarrow A\ true}{\Delta \Longrightarrow A\ lax}\ lax_R$$

- Lax truth does not entail truth
- Judgmental definition: (always silently expanded on the left)

$$\left[\ \frac{\Delta, A\ res \Longrightarrow C\ lax}{\Delta, A\ lax \Longrightarrow C\ lax}\ lax_L\ \right]$$

- Applies only with lax succedent, not truth

Judgmental Principles

- No new identity principle

$$\dfrac{\dfrac{\overline{A\ \mathit{res} \Longrightarrow A\ \mathit{true}}\ \mathrm{id}_A}{A\ \mathit{res} \Longrightarrow A\ \mathit{lax}}\ \mathit{lax}_R}{A\ \mathit{lax} \Longrightarrow A\ \mathit{lax}}\ \mathit{lax}_L$$

- Derived cut principle

$$\dfrac{\Delta \Longrightarrow A\ \mathit{lax} \quad \dfrac{\Delta', A\ \mathit{res} \Longrightarrow C\ \mathit{lax}}{\Delta', A\ \mathit{lax} \Longrightarrow C\ \mathit{lax}}\ \mathit{lax}_R}{\Delta, \Delta' \Longrightarrow C\ \mathit{lax}}\ \mathrm{cut}_{\mathit{lax}}$$

- Allow $\gamma ::= C\ \mathit{true} \mid C\ \mathit{lax}$ in all other rules with generic succedent

Lax Modality = Effect Monad

- Internalize lax judgment as proposition $\{A\}$

$$\frac{\Delta \Longrightarrow A \; lax}{\Delta \Longrightarrow \{A\} \; true}\,\{\,\}R \qquad\qquad \frac{\Delta, A \; res \Longrightarrow C \; lax}{\Delta, \{A\} \; res \Longrightarrow C \; lax}\,\{\,\}L$$

- Identity expansion

$$\frac{}{\{A\} \Longrightarrow \{A\}}\,\mathrm{id}_{\{A\}} \quad \longrightarrow_E \quad \dfrac{\dfrac{\dfrac{\dfrac{}{A \Longrightarrow A}\,\mathrm{id}_A}{A \Longrightarrow A \; lax}\,lax_L}{\{A\} \Longrightarrow A \; lax}\,\{\,\}L}{\{A\} \Longrightarrow \{A\}}\,\{\,\}R$$

Cut Reduction for Lax Modality

$$
\dfrac{\dfrac{\overset{\mathcal{D}}{\Delta \Longrightarrow A\ lax}}{\Delta \Longrightarrow \{A\}}\ \{\,\}R \qquad \dfrac{\overset{\mathcal{E}}{\Delta' \Longrightarrow A \Longrightarrow C\ lax}}{\Delta', \{A\} \Longrightarrow C\ lax}\ \{\,\}L}{\Delta, \Delta' \Longrightarrow C\ lax}\ \mathrm{cut}_{\{A\}}
$$

$$
\longrightarrow_R \qquad \dfrac{\overset{\mathcal{D}}{\Delta \Longrightarrow A\ lax} \qquad \overset{\mathcal{E}}{\Delta', A \Longrightarrow C\ lax}}{\Delta, \Delta' \Longrightarrow C\ lax}\ \mathrm{cut}_{A\ lax}
$$

Outline

1. Background: Proof-Carrying Authorization
2. Logical Foundations
    1. Resources (linear logic)
    2. Possessions (linear epistemic logic)
    3. Effects (linear lax logic)
    4. From axioms to inference rules via focusing
    5. Persistent truth and knowledge (epistemic logic)
3. Policy Consequences
    1. State invariants
    2. Proving metatheorems
4. Speculation: linear epistemic logic programming

Polarization

- Focusing: we can obtain a complete big-step proof system using two observations
  - Apply invertible rules eagerly
  - When all top-level propositions have non-invertible rules, focus on one of them and apply a run of non-invertible rules to its components
- Robust technique (all reasonable known logics?)
- Polarization: we explicitly categorize propositions into negative (invertible right) and positive (invertible left).
  - Here: exploit monad (other choices are possible)

$$
\begin{array}{lccl}
\text{Negative} & A^- & ::= & P^- \mid A^+ \multimap A^- \mid \{A^+\} \\
\text{Positive} & A^+ & ::= & A_1 \otimes A_2 \mid \mathbf{1} \mid [K]A^- \mid A^-
\end{array}
$$

Example: Focusing

Write A for formula in focus
Must apply rule to focus formula

$$
\dfrac{\dfrac{\dfrac{\dfrac{\Delta, \mathit{fp}\ \textbf{has}\ \text{coffee}, \mathit{tdo}\ \textbf{has}\ \$ \Longrightarrow C\ \mathit{lax}}{\Delta, \mathit{fp}\ \textbf{has}\ \text{coffee}, [\mathit{tdo}]\$ \Longrightarrow C\ \mathit{lax}}\ [\,]L}{\Delta, [\mathit{fp}]\mathit{coffee}, [\mathit{tdo}]\$ \Longrightarrow C\ \mathit{lax}}\ [\,]L}{\Delta, [\mathit{fp}]\text{coffee} \otimes [\mathit{tdo}]\$ \Longrightarrow C\ \mathit{lax}}\ \otimes L}{\Delta, \{[\mathit{fp}]\text{coffee} \otimes [\mathit{tdo}]\$\} \Longrightarrow C\ \mathit{lax}}\ \{\,\}L
$$

$$
\dfrac{\dfrac{\dfrac{\dfrac{}{\$ \Longrightarrow \$}\ \text{id}}{\mathit{fp}\ \textbf{has}\ \$ \Longrightarrow \$}\ \textbf{has}_L}{\mathit{fp}\ \textbf{has}\ \$ \Longrightarrow [\mathit{fp}]\$}\ [\,]R \quad \dfrac{\dfrac{\dfrac{}{\text{beans} \Longrightarrow \text{beans}}\ \text{id}}{\mathit{tdo}\ \textbf{has}\ \text{beans} \Longrightarrow \text{beans}}\ \textbf{has}_L}{\mathit{tdo}\ \textbf{has}\ \text{beans} \Longrightarrow [\mathit{tdo}]\text{beans}}\ [\,]R}{\mathit{fp}\ \textbf{has}\ \$, \mathit{tdo}\ \textbf{has}\ \text{beans} \Longrightarrow [\mathit{fp}]\$ \otimes [\mathit{tdo}]\text{beans}}\ \otimes R \qquad \text{see above}
$$

$$
\overline{\Delta, \mathit{fp}\ \textbf{has}\ \$, \mathit{tdo}\ \textbf{has}\ \text{beans}, [\mathit{fp}]\$ \otimes [\mathit{tdo}]\text{beans} \multimap \{[\mathit{fp}]\text{coffee} \otimes [\mathit{tdo}]\$\} \Longrightarrow C\ \mathit{lax}}\ \multimap L
$$

From Axioms to Inference Rules

- Focusing allows us to turn axioms such as

$$\text{buy} : [\mathit{fp}]\$ \otimes [\mathit{tdo}]\text{beans} \multimap \{[\mathit{fp}]\text{coffee} \otimes [\mathit{tdo}]\$\}$$

  into a complete set of derived inference rules such as

$$\frac{\Delta, \mathit{fp}\ \textbf{has}\ \text{coffee}, \mathit{tdo}\ \textbf{has}\ \$ \Longrightarrow C\ \mathit{lax}}{\Delta, \mathit{fp}\ \textbf{has}\ \$, \mathit{tdo}\ \textbf{has}\ \text{beans} \Longrightarrow C\ \mathit{lax}}\ \text{buy}$$

- Aside: to get this specific rule, some assumption on K's possessions and other axioms are necessary
  - No axioms with "head" \$
  - Possessions are of the form K **has** P for atoms P
- The lax modality allows for somewhat stricter proof control than just focusing

Example Revisited: Deleting a File

Key to Syntax
$\langle K\rangle A$ = "K **says** A"
$[K]A$ = "K **has** A"
$[\![K]\!]A$ = "K **knows** A"
$\{A\}$ = "A, with effect"

$$
\begin{array}{ll}
 & \langle K\rangle \mathrm{do}(K, \mathrm{create}) \\
\otimes & \langle fs\rangle \mathrm{may}(K, \mathrm{create}) \\
\multimap & \{\exists f\,.\,\exists v. \\
 & \qquad\quad !\langle fs\rangle \mathrm{owns}(K, f) \\
 & \qquad \otimes\ [fs]\mathrm{current}(f, v) \\
 & \qquad \otimes\ [\![fs]\!]\mathrm{contains}(f, v, \text{""}) \\
 & \qquad \otimes\ [\![K]\!]\mathrm{contains}(f, v, \text{""})\}
\end{array}
$$

- To explain: knowledge $[\![K]\!]A$ and persistent truth $!A$
- Following our judgmental approach, we add new form of assumptions

Persistent Assumptions

- Sequents have form

$$\Gamma; \Delta \Longrightarrow \gamma$$

where

$$\begin{array}{lcll} \text{Persistent ants.} & \Gamma & ::= & \bullet \mid \Gamma, A\ \mathit{pers} \mid \Gamma, K\ \textbf{knows}\ A \\ \text{Linear ants.} & \Delta & ::= & \bullet \mid \Delta, A\ \mathit{res} \mid \Delta, K\ \textbf{has}\ A \\ \text{Succedents} & \gamma & ::= & A\ \mathit{true} \mid A\ \mathit{lax} \end{array}$$

- Persistent assumptions grow monotonically in bottom-up proof construction
- All present rules are updated to propagate Γ to all premises

Persistent Truth

- Persistent truths can be used

$$\frac{A\,pers \in \Gamma \quad \Gamma; \Delta, A\,res \Longrightarrow \gamma}{\Gamma; \Delta \Longrightarrow \gamma}\ pers_L$$

- Truths whose proof requires no consumable resources are persistent

$$\left[\ \frac{\Gamma; \bullet \Longrightarrow A\ true}{\Gamma; \bullet \Longrightarrow A\ pers}\ pers_R\ \right]$$

(The rule line in $pers_R$ is printed as a double line.)

Cut and Identity for Persistent Truth

- No new identity principle

$$\dfrac{\dfrac{\overline{A\ pers; A\ res \Longrightarrow A\ true}\ \text{id}}{A\ pers; \bullet \Longrightarrow A\ true}\ pers_L}{A\ pers; \bullet \Longrightarrow A\ pers}\ pers_R$$

- New derived cut principle

$$\dfrac{\dfrac{\Gamma; \bullet \Longrightarrow A\ true}{\Gamma; \bullet \Longrightarrow A\ pers}\ pers_R \qquad \Gamma, A\ pers; \Delta \Longrightarrow \gamma}{\Gamma; \Delta \Longrightarrow \gamma}\ \text{cut}_{pers}$$

The Exponential Modality of Linear Logic

$$\frac{\Gamma; \bullet \Longrightarrow A\ \mathit{true}}{\Gamma; \bullet \Longrightarrow !A\ \mathit{true}}\ !R \qquad \frac{\Gamma, A\ \mathit{pers}; \Delta \Longrightarrow \gamma}{\Gamma; \Delta, !A\ \mathit{res} \Longrightarrow \gamma}\ !L$$

- Internalize persistent truth
- Identity expansion and cut reduction work easily

A Judgment of Knowledge

- K **knows** $A \sim$ knowledge as persistent possession
- Persistent knowledge can be used by K

$$\frac{K\ \mathbf{knows}\ A \in \Gamma \quad \Gamma; \Delta, A\ \mathit{res} \Longrightarrow \gamma}{\Gamma; \Delta \Longrightarrow \gamma}\ \mathbf{knows}_L$$

- Truth whose proofs require only local knowledge can be known

$$\left[\ \frac{\Gamma|_K; \bullet \Longrightarrow A}{\Gamma; \bullet \Longrightarrow K\ \mathbf{knows}\ A}\ \mathbf{knows}_R\ \right]$$

- $\Gamma|_K$ restricts to antecedents of the form K **knows** _

Cut and Identity for Knowledge

- No new identity principle

$$\dfrac{\dfrac{\overline{K\ \mathbf{knows}\ A; A\ \mathit{res} \Longrightarrow A\ \mathit{true}}\ \mathsf{id}}{K\ \mathbf{knows}\ A; \bullet \Longrightarrow A\ \mathit{true}}\ \mathbf{knows}_L}{K\ \mathbf{knows}\ A; \bullet \Longrightarrow K\ \mathbf{knows}\ A}\ \mathbf{knows}_R$$

- New derived cut principle

$$\dfrac{\dfrac{\Gamma|_K; \bullet \Longrightarrow A\ \mathit{true}}{\Gamma; \bullet \Longrightarrow K\ \mathbf{knows}\ A}\ \mathbf{knows}_R \qquad \Gamma, K\ \mathbf{knows}\ A; \Delta \Longrightarrow \gamma}{\Gamma; \Delta \Longrightarrow \gamma}\ \mathsf{cut}_{\mathbf{knows}}$$

Knowledge as a Modality

$$\frac{\Gamma|_K ; \bullet \Longrightarrow A\ \mathit{true}}{\Gamma ; \bullet \Longrightarrow [\![K]\!]A\ \mathit{true}}\ [\![\ ]\!]R \qquad \frac{\Gamma, K\ \textbf{knows}\ A ; \Delta \Longrightarrow \gamma}{\Gamma ; \Delta, [\![K]\!]A \Longrightarrow \gamma}\ [\![\ ]\!]L$$

- Identity expansion and cut reduction as usual
- Knowledge is like indexed judgmental S4

Outline

Characterizing State

- Need to characterize the system states so we can reason about the policy
- System states are pairs $\Gamma; \Delta$
  - Γ is persistent
  - Δ is linear
  - We do not care about the right-hand side, but it must have the form C *lax* to permit effects
- Using this characterization, we turn each semantics rule into (one ore more) rewrite rules for system states
- Using the rewrite rules we can prove theorems about the semantics

Example: Characterizing File System State

- Each persistent judgment in Γ is one of
  - A policy rule or semantics action
  - fs **knows** contents(F, V, S) or
    K **knows** contents(F, V, S)
  - $\langle \mathit{fs} \rangle$user$(K)$ or $\langle \mathit{fs} \rangle$owns$(K, F)$
- Each linear judgment in Δ is one of
  - fs **has** current(F, V)
  - $\langle K \rangle$do(K, A)
- For each file F, there is at most one V such that fs **has** current(F, V)

Example: Reading a File

- Specification

$$\begin{array}{ll} & \langle K\rangle \mathsf{do}(K, \mathsf{on}(F, \mathsf{delete})) \\ \otimes & \langle fs\rangle \mathsf{may}(K, \mathsf{on}(F, \mathsf{delete})) \\ \otimes & [fs]\mathsf{current}(F, V) \\ \multimap & \{\mathbf{1}\} \end{array}$$

- Rewrite step

$$\begin{array}{ll} & \Gamma; \Delta, \langle K\rangle \mathsf{do}(K, \mathsf{on}(F, \mathsf{delete})), fs\ \mathbf{has}\ \mathsf{current}(F, V) \\ \rightarrow & \Gamma; \Delta \end{array}$$

provided $\Gamma \vdash \langle fs\rangle \mathsf{may}(K, \mathsf{on}(F, \mathsf{delete}))$

Example: Writing to a File

- Specification

$$
\begin{aligned}
& \langle K \rangle \mathrm{do}(K, \mathrm{on}(F, \mathrm{write}(S))) \\
\otimes\ & \langle \mathit{fs} \rangle \mathrm{may}(K, \mathrm{on}(F, \mathrm{write}(S))) \\
\otimes\ & [\mathit{fs}]\mathrm{current}(F, V) \\
\multimap\ & \{\exists v'.\, [\mathit{fs}]\mathrm{current}(F, v') \\
& \quad \otimes [\![\mathit{fs}]\!]\mathrm{contains}(F, v', S) \\
& \quad \otimes [\![K]\!]\mathrm{contains}(F, v', S)\}]
\end{aligned}
$$

- Rewrite interpretation

$$
\begin{aligned}
& \Gamma; \Delta, \langle K \rangle \mathrm{do}(K, \mathrm{on}(F, \mathrm{write}(S))), \mathit{fs}\ \mathbf{has}\ \mathrm{current}(F, V) \\
\rightarrow\ & \Gamma, \mathit{fs}\ \mathbf{knows}\ \mathrm{contains}(F, v', S), K\ \mathbf{knows}\ \mathrm{contains}(F, v', S); \\
& \Delta, \mathit{fs}\ \mathbf{has}\ \mathrm{current}(F, v')
\end{aligned}
$$

for a new v' provided $\Gamma \vdash \langle \mathit{fs} \rangle \mathrm{may}(K, \mathrm{on}(F, \mathrm{write}(S)))$

Analysis Example: Policy Controls Knowledge

Theorem (Knowledge Safety)

If $\Gamma; \Delta$ *is a file system state such that*

$$\Gamma; \Delta \longrightarrow \Gamma', K \textbf{ knows } \mathrm{contents}(F, V, S); \Delta'$$

then either K **knows** $\mathrm{contents}(F, T, S) \in \Gamma$ *or the step was a create, read, or write action* A *on* F *by* K *permitted by the policy (as evidenced by a proof of* $\langle fs \rangle \mathrm{may}(K, A)$*)*

Proof.

By case analysis of the possible rewrite step schemata. □

Stratification

- The proofs still apply as long as the signed policy statements do not involve any effects or possessions
- In general, the system should be stratified so proofs of authorization are effect-free
  - Uses of authorizations are the effect
  - Linear theorem proving of authorization theorem does not consume the certificates!
- Located certificates and proofs
  - File system example abstract away from location of proofs
  - Could specify client of server to produce the proof

Another Example: Electronic Voting

va = voting authority

$$
\begin{array}{ll}
\quad \langle va\rangle \mathrm{hasvote}(K) & \text{(linear certificate)} \\
\otimes\ !\langle va\rangle \mathrm{candidate}(L) & \text{(persistent certificate)} \\
\otimes\ [K]\langle K\rangle \mathrm{votefor}(L) & \text{(linear possession of cert.)} \\
\otimes\ [va]\mathrm{voting} & \text{(linear ``token'')} \\
\otimes\ [va]\mathrm{votecount}(N) & \text{(linear ``token'')} \\
\multimap \{[va]\mathrm{vote}(L) & \text{(linear vote result)} \\
\qquad \otimes\ [va]\mathrm{count}(N+1) & \\
\qquad \otimes\ [va]\mathrm{voting}\} &
\end{array}
$$

Example: Counting Electronic Votes

$$
\begin{array}{lll}
 & [va]\mathsf{voting} & \text{(linear "token")} \\
\otimes & \langle va\rangle\mathsf{pollclosed} & \text{(linear trigger)} \\
\otimes & [va]\mathsf{votecount}(N) & \text{(linear "token")} \\
\multimap & \{[va]\mathsf{counting}(N)\} & \text{(new token)} \\
 & & \\
 & [va]\mathsf{counting}(0) & \text{(vote counting done)} \\
\multimap & \{[va]\mathsf{done}\} & \\
 & & \\
 & [va]\mathsf{counting}(N) \otimes !N > 0 & \text{(token and condition)} \\
\otimes & [va]\mathsf{votefor}(L) & \text{(vote for } L\text{, being tallied)} \\
\otimes & [va]\mathsf{numvotes}(L, K) & \text{(vote counter)} \\
\multimap & \{[va]\mathsf{counting}(N) & \\
 & \quad \otimes [va]\mathsf{votes}(L, K+1)\} &
\end{array}
$$

Outline

1. Background: Proof-Carrying Authorization
2. Logical Foundations
   1. Resources (linear logic)
   2. Possessions (linear epistemic logic)
   3. Effects (linear lax logic)
   4. From axioms to inference rules via focusing
   5. Persistent truth and knowledge (epistemic logic)
3. Policy Consequences
   1. State invariants
   2. Proving metatheorems
4. Speculation: linear epistemic logic programming

Speculation: Linear Epistemic Logic Programming

- Idea: Give a forward chaining ("bottom-up") logic programming interpretation as a distributed programming language
- By design, the implementation will satisfy the specification
- By design, the implementation will satisfy the theorems proven about the specification
- Based on the polarized, focusing interpretation
  - Some additional restrictions will be necessary
  - Mode checking, staging verification, ...
- Must execute protocols on multiple hosts

Example: A Binary Counter

- State invariants for each principal (= bit) K
  - For each K, either K **knows** next(L) or K **knows** last
  - For each K, either K **has** zero or K **has** one
  - For one K, K **has** inc may be present
- Program

$$[K]\mathrm{inc} \otimes [K]\mathrm{zero} \multimap \{[K]\mathrm{one}\}$$

$$[K]\mathrm{inc} \otimes [K]\mathrm{one} \otimes [\![K]\!]\mathrm{next}(L) \multimap \{[K]\mathrm{zero} \otimes [L]\mathrm{inc}\}$$

$$[K]\mathrm{inc} \otimes [K]\mathrm{one} \otimes [\![K]\!]\mathrm{last} \multimap \{[K]\mathrm{zero}\}$$

- Have hand-compiled version in Meld on "blinky-blocks"

Atomicity

- In general, complex multi-party contract signing protocols may be necessary to ensure atomicity of the rules
- Example (with conditions from two parties)

$$[\![L]\!]\mathsf{prev}(K) \otimes [K]\mathsf{carry} \otimes [L]\mathsf{zero} \multimap \{[L]\mathsf{one}\}$$

$$[\![L]\!]\mathsf{prev}(K) \otimes [K]\mathsf{carry} \otimes [L]\mathsf{one} \multimap \{[L]\mathsf{zero} \otimes [L]\mathsf{carry}\}$$

- Inference system suggests "truth" as a trusted third party that leaks no information
- Looking for a suitable lower-level calculus to compile to for expressing communication protocols

Summary

- Goals
  - Logical specification of distributed authorization policies
  - Reliable enforcement of such high-level policies (PCA)
    - Implemented in practical proof-carrying file system
  - Mechanized reasoning about consequences of policies:
    - Evolution of system state
    - Principals' knowledge (information)
    - Principals' possessions (consumable resources)
- Approach: linear epistemic logic
  - Pedantic definition from judgmental principles
  - Possession is linear knowledge
  - Specification at extremely high level of abstraction

Ongoing and Future Work

- Define distributed forward chaining linear epistemic logic programming language
- Compile to distributed code executing multi-party communication protocols
- Prove correctness with respect to rewriting semantics
  - Atomicity of rules most difficult
  - Identify tractable language subset
  - Eliminate some uses of trusted third party (= truth)
- Mechanize reasoning about policies
  - "See" my talk at LFMTP yesterday

For More . . .

- H. DeYoung and F. Pfenning, *Reasoning about the Consequences of Authorization Policies in a Linear Epistemic Logic*, Workshop on Foundations of Computer Security (FCS), 2009.
- D. Garg et al., *A Linear Logic of Affirmation and Knowledge*, European Symposium on Research in Computer Security (ESORICS), 2006.
- Further pointers from this workshop, I hope!